BAB II

LANDASAN TEORI

Landasan teori dalam penyusunan tugas akhir ini menggunakan beberapa teori pendukung yang akan digunakan untuk menentukan lintasan terpendek pada jarak Desa di Kecamatan Rengat Barat.

2.1 Graf

a. Definisi Graf

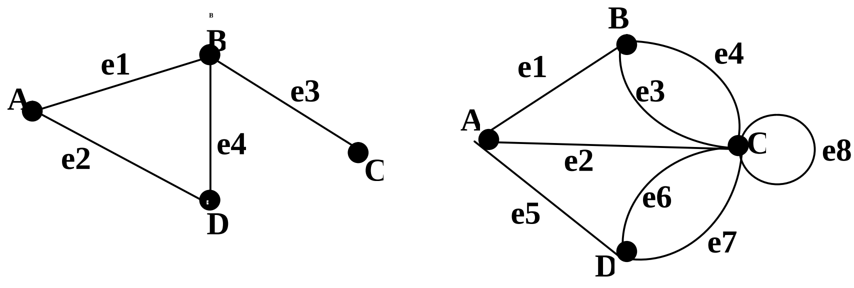
Siang (2006) mengatakan graf adalah suatu graf G terdiri dari 2 himpunan yang berhingga, yaitu himpunan titik-titik tidak kosong (simbol $V(G)$) dan himpunan garis-garis (simbol $E(G)$).

Munir (2007) graf G didefinisikan sebagai pasangan himpunan (V, E), ditulis dengan notasi G = (V, E) yang dalam hal ini V adalah himpunan tidak kosong dari simpul-simpul (*vertices* atau *node*) dan E adalah himpunan sisi (*edge* atau *arcs*) yang dihubungkan sepasang simpul.

Berdasarkan dari dua definisi tersebut dapat dinyatakan bahwa V tidak boleh kosong sedangkan E boleh kosong, jadi sebuah graf dimungkinkan tidak memiliki sisi satu buah pun, tetapi simpulnya harus ada minimal satu. Graf yang hanya memiliki satu titik tanpa sisi disebut graf trivial.

Setiap garis berhubungan dengan satu atau dua titik. Titik-titik tersebut dinamakan titik ujung. Garis yang hanya berhubungan dengan satu titik ujung disebut loop. Dua garis yang berbeda yang manghubungkan titik yang sama disebut garis paralel. Dua titik dikatakan berhubungan jika ada garis yang menghubungkan keduanya. Titik yang tidak memiliki garis yang berhubungan dengannya disebut titik terasing.

Suatu graf ditulis dengan notasi $G = (V, E)$. Gambar berikut ini merupakan contoh graf.



G_1 G_2

Gambar 2.1 Graf

Berdasarkan Gambar 1, graf $G_1 = (V, E)$ di atas yaitu:

$V = \{A, B, C, D\}$

$E = \{\{A, B\}, \{A, D\}, \{B, C\}, \{B, D\}\}$

$= \{e_1, e_2, e_3, e_4\}$

Gambar 1 Graf $G_2 = (V, E)$ yaitu:

$V = \{A, B, C, D\}$

$E = \{\{A, B\}, \{A, C\}, \{B, C\}, \{B, C\}, \{A, D\}, \{D, C\}, \{D, C\}, \{C, C\}\}$

$= \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8\}$

b. Jenis-jenis Graf

Berdasarkan ada tidaknya gelang atau sisi ganda pada suatu graf, maka graf digolongkan menjadi dua jenis:

1. Graf Sederhana (*Simple Graph*)

   Graf yang tidak memiliki gelang atau sisi ganda. G_1 pada Gambar 1 adalah contoh graf sederhana.

2. Graf Tak-Sederhana (*Unsimple Graph*)

   Graf yang memiliki sisi ganda atau gelang.ada dua jenis graf tak-sederhana, yaitu graf ganda (*multigraph*) dan graf semu (*pseudograph*). Graf ganda adalah graf yang mengandung sisi ganda, sisi ganda yang menghubungkan sepasang simpul bisa lebih dari dua buah. Graf semu adalah graf yang mengandung gelang (*loop*). Graf semu lebih umum daripada graf ganda, karena sisi pada graf semu dapat terhubung kedirinya sendiri. G_2 pada gambar 1 merupakan contoh graf tak-sederhana.

Berdasarkan jumlah simpul pada suatu graf, maka graf dapat digolongkan menjadi:

1. Graf Berhingga (*Limited Graph*)

   Graf berhingga adalah graf yang jumlah simpulnya n berhingga.

2. Graf Tak-Berhingga (*Unlimited Graph*)

   Yaitu graf yang simpulnya n tak-berhingga banyaknya.

Berdasarkan orientasi arah pada sisinya, graf dibedakan menjadi:

1. Graf Tak-Berarah (*Undirected Graph*)

   Yaitu graf yang sisinya tidak mempunyai orientasi arah. Pada graf tak berarah ururtan pasangan simpul yang dihubungkan oleh sisi tidak diperhatikan. Jadi, u, v = v, u adalah sisi yang sama. Contohnya pada Gambar 1.

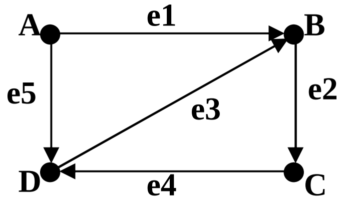
2. Graf Berarah (*Directed Graph* atau *Digraph*)

   Yaitu graf yang setiap sisinya diberikan orientasi arah atau biasanya disebut dengan busur (*arc*). Pada graf berarah u, v dan v, u menyatakan dua buah busur yang berbeda, dengan kata lain $u, v \neq v, u$. Untuk busur u, v , simpul u dinamakan simpul asal (*initial vetex*) dan simpul v dinamakan simpul terminal (*terminal vertex*).

Contoh 2.1:

Gambarlah sebuah graf berarah dengan ketentuan sebagai berikut: e_1 = A, B , e_2 = B, C , e_3 = D, B , e_4 = C, D , e_5 = A, D

Penyelesaian:



Gambar 2.2 Graf Berarah

c. **Terminologi Graf**

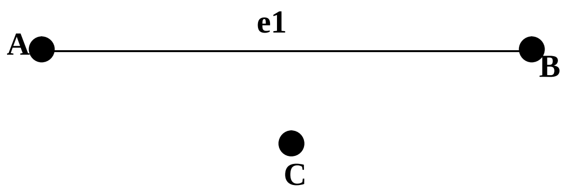
Terminologi (istilah) yang berkaitan dengan graf adalah sebagai berikut:

1. Bertetangga (*Adjacent*)

Didefinisikan bertetangga dan bersisian dalam sebuah graf. Misalkan $e = \{(u, v)\}$ adalah sebuah sisi dalam G, yaitu u dan v adalah titik-titik ujung dari e. simpul u dikatakan bertetangga terhadap simpul v dan sisi e dikatakan bersisian atau terhubung dengan u dan v. (Lipschuts, 2002)

2. Simpul Terpencil (*Isolated Vertex*)

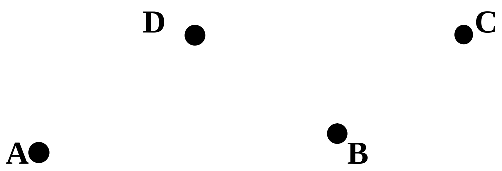
Munir (2007) simpul terpencil adalah simpul yang tidak mempunyai sisi yang bersisian dengannya atau dapat dinyatakan bahwa simpul terpencil adalah simpul yang tidak satupun bertetangga dengan simpul-simpul lainnya. Gambar berikut adalah contoh graf terpencil.



Gambar 2.3 Simpul Terpencil

3. Graf Kosong (*Null Graph* atau *Empty Graph*)

Graf *Null* atau graf kosong yaitu graf dengan n titik, dinotasikan N_n yang dalam hal ini n adalah jumlah simpul, graf kosong didefinisikan sebagai suatu graf dengan himpunan sisinya merupakan himpunan kosong. Graf ini hanya terdiri dari himpunan elemen yang disebut vertex. Gambar berikut merupakan graf kosong N_4.



Gambar 2.4 Graf Kosong N_4

4. Derajat (*Degree*)

Misalkan v adalah titik dalam suatu graf G. Derajat titik v (simbol $d(v)$) adalah jumlah garis yang berhubungan dengan titik v dan garis suatu loop dihitung dua kali. Derajat total G adalah jumlah derajat semua titik dalam G.

5. Graf Terhubung (*Connected Graph*)

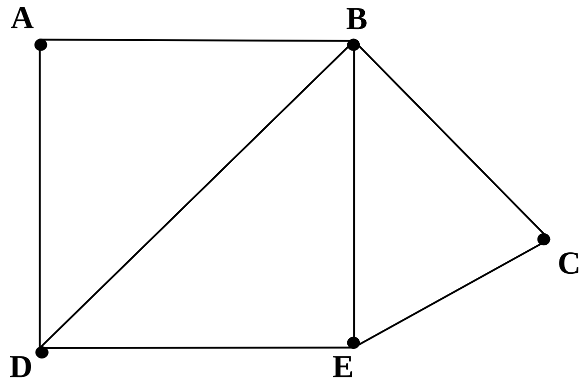
Dua simpul v dan w dalam G dikatakan terhubung jika dan hanya jika ada lintasan dari simpul v ke w atau w ke v .(Siang, 2006)

6. Lintasan (*Path*)

Lintasan yang panjangnya n dari simpul awal v_0 ke simpul tujuan didalam graf G ialah barisan berselang-seling simpul-simpul dan sisi-sisi yang berbentuk $v_0, e_1, v_1, e_2, \dots, v_{n-1}, e_n, v_n$ maka e_1 = v_0, v , e_2 v_1, v_2 , ... , e_n v_{n-1}, v_n adalah sisi-sisi dari graf G.

Contoh 2.2:

Graf berikut ini merupakan gambaran suatu daerah:



Gambar 2.5 Lintasan pada Graf

Berdasarkan Gambar 4 di atas dimisalkan bahwa simpul-simpul dalam graf tersebut mewakili kota, sisi mewakili jalan antara dua kota. Misalkan seseorang berada di kota A dan ingin berpergian ke kota E. Maka akan didapatkan beberapa lintasan yang dapat ditempuh.

Penyelesaian:

1. p_1 = A, B, E
2. p_2 = A, B, C, E
3. p_3 = A, B, D, E
4. p_4 = A, D, E

7. Siklus (*Cycle*) atau Sirkuit (*Circuit*)

Sirkuit adalah lintasan tertutup yang berawal dan berakhir pada simpul yang sama. Panjang dari sebuah sirkuit dihitung dari berapa banyak sisi yang terdapat pada sirkuit tersebut.

8. Graf Berbobot

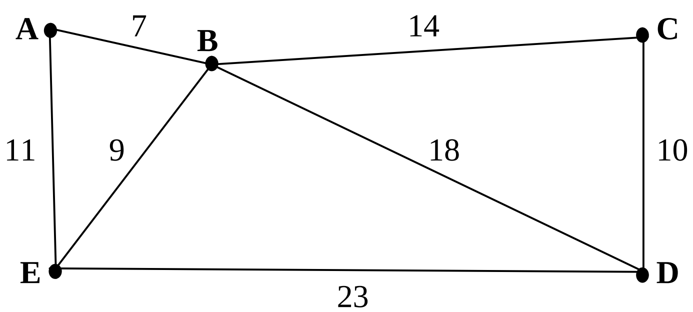
Bila sisi e dalam graf G dikaitkan dengan sebuah bilangan real $w\ e$, maka $w\ e$ disebut bobot (weight) dari e. Bobot dari sebuah graf G, dinotasikan dengan $w\ G$.

Contoh 2.3:

Gambarlah sebuah graf berbobot dengan ketentuan sebagai berikut: A, B = 7,

B, C = 14, C, D = 10, B, D = 18, B, E = 9, A, E = 11, D, E = 23

Penyelesaian:



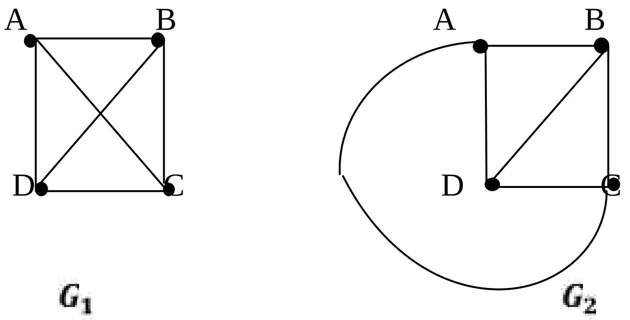
Gambar 2.6 Graf Berbobot

9. Graf Planar dan Graf Bidang

Sebuah graf disebut graf planar jika graf tersebut dapat digambar pada bidang datar sedemikian sehingga sisi-sisinya hanya beririsan (berpotongan) simpul-simpul akhirnya. (Zulkarnain, 2006)

Graf bidang adalah graf planar G yang digambarkan pada bidang datar sedemikian hingga tidak ada sisi-sisinya yang saling beririsan (berpotongan) kecuali pada simpul-simpul akhir sisi-sisi tersebut. (Zulkarnain, 2006)

Berikut adalah



Gambar 2.7 (G_1) Graf Planar dan (G_2) Graf Bidang

Berdasarkan Gambar 4 di atas, dapat dilihat bahwa graf pada G_1 yang awalnya terdapat sisi yang saling berpotongan yaitu $e = \ A, C$ saling berpotongan dengan $e = \ B, D$ dapat digambarkan kembali seperti G_2, sehingga tidak ada sisi yang berpotongan.

10. Graf Dual

Secara geometri graf dual adalah sebuah graf planar G yang direpresentasikan sebagai graf bidang dan dapat dibuat suatu graf G^*. Aplikasi dari graf dual adalah untuk mempresentasikan peta, karena setiap peta pada bidang datar terdiri dari sejumlah wilayah (Munir,2005).

Cara membuat graf dual dari peta (Lipschuts, 2002) :

a) Setiap wilayah pada peta buat sebuah titik
b) Jika dua wilayah mempunyai sebuah sisi bersama maka simpul-simpul yang terkait dapat dihubungkan dengan sebuah garis melalui sisi bersama tersebut.

d. Representasi Graf dalam Matriks

Suatu graf dapat dinyatakan menggunakan matriks. Berikut adalah representasi graf dalam matriks:

1. Matriks Ketetanggaan (*Adjacency Matrix*)

Matriks ketetanggaan digunakan untuk menyatakan keterhubungan antar simpul-simpulnya. Jumlah baris dan kolom pada matriks ketetanggaan sama dengan jumlah simpul dalam graf atau graf bujur sangkar $n \times n, A = (a_{ij})$ dengan jika ada sisi dari simpul v_i ke v_j maka elemen matriksnya bernilai 1, dan sebaliknya maka elemen matriks bernilai 0.

2. Matriks Bersisian (*Incidency Matrix*)

Matriks bersisian digunakan untuk manyatakan kebersisian simpul dengan sisi. Matriks bersisian adalah matriks yang berukuran $n \times m$. Baris menunjukan label simpul dan kolom menunjukan label sisinya. Jika simpul terhubung dengan sisi maka elemen matriks bernilai 1, sebaliknya jika simpul tidak terhubung dengan sisi, maka elemen matriks bernilai 0.

e. Lintasan Terpendek

Lintasan terpendek merupakan lintasan minimum yang diperlukan untuk mencapai suatu tempat dari tempat tertentu. Lintasan yang dimaksud tersebut dapat dicari dengan menggunakan graf. Persoalan dalam mencari lintasan terpendek ini sering terjadi dalam kehidupan sehari-hari. Graf yang digunakan dalam pencarian lintasan terpendek adalah graf berbobot (*weight graph*), yaitu graf yang setiap sisinya diberikan suatu nilai atau bobot. Bobot pada sisi graf dapat menyatakan jarak antar kota, waktu pengiriman pesan, ongkos pembangunan, dan sebagainya. Asumsi yang digunakan adalah bahwa semua bobot bernilai positif. Kata terpendek berarti meminimisasi bobot pada suatu lintasan di dalam graf.

Terdapat beberapa jenis persoalan lintasan terpendek, antara lain:

1. Lintasan terpendek antara dua simpul tertentu.
2. Lintasan terpendek antara semua pasangan simpul.
3. Lintasan terpendek dari simpul tertentu ke semua simpul yang lain.

4. Lintasan terpendek antara dua buah simpul yang melalui beberapa simpul tertentu

2.2 Algoritma Floyd-Warshall

Algoritma Floyd Warshall adalah algoritma yang ditemukan oleh Stephen Warshall dan Robert W. Floyd. Stephen Warshall lahir di New York pada tahun 1935 dan meninggal pada tanggal 11 desember 2006. Robert W. Floyd, lahir di New York pada tanggal 8 juni 1936 dan meninggal pada tangggal 25 September 2001.

Algoritma Warshall untuk mencari lintasan terpendek dan merupakan algoritma yang sederhana dan mudah implementasinya. Algoritma ini dalam menentukan lintasan terpendek dengan cara memulai iterasi dari titik awalnya kemudian memperpanjang lintasan dengan mengevaluasi titik demi titik hingga mencapai titik tujuan dengan jumlah lintasan yang sekecil mungkin.

Dalam itersinya mencari lintasan terpendek, algoritma ini akan membentuk n matriks, sesuai dengan iterasi-k. Hal ini menyebabkan waktu yang lambat dalam mencari lintasan terpendeknya jika n yang besar. Tetapi, meskipun agak lambat cara proses mancarinya algoritma ini banyak digunakan juga karena kesederhanaan algoritmanya, dan implemantasinya yang sangat mudah.

2.3 Algoritma Bellman-Ford

Algoritma Bellman-Ford yang ditemukan oleh Richard E. Bellman, seorang ahli matematika yang terlahir di New York 1920. Algoritma Bellman-Ford menghitung jarak terpendek (dari satu sumber) pada sebuah graf berbobot. Maksudnya dari satu sumber ialah bahwa ia menghitung semua jarak terpendek yang berawal dari satu titik node. Algoritma Dijkstra dapat lebih cepat mencari hal yang sama dengan syarat tidak ada sisi (edge) yang berbobot negatif. Maka Algoritma Bellman-Ford hanya digunakan jika ada sisi berbobot negatif. Muncul nya Algoritma ini cukup membantu jika bobot dari suatu graf bernilai negatif.

Langkah-langkah untuk menyelesaikan lintasan terpendek dengan menggunakan Algoritma Bellman-ford adalah:

a. Menentukan vertek asal dan vertek tujuan.
b. Memberikan tanda + atau – pada bobot graf.
c. Dimulai dari vertek “0” yang menyebarkan informasi ke masing-masing vertek yang langsung berhubungan dengan vertek “0” tersebut sesuai dengan nilai bobot jalurnya. Semua vertek yang sudah memiliki nilai akan menyebarkan informasi ke vertek yang saling berhubungan langsung kemudian jika sebuah vertek diisi oleh lebih dari satu nilai maka nilai yang dipilih adalah nilai yang terkecil.
d. Lakukan berulang kali langkah c sampai terbentuknya nilai pada setiap vertek yang sudah terjelajah